### תיאור קצר של הפיצ'רים שבחרנו לממש בתרגיל הקודם:

* רשימת משאלות- המשתמש יכול להוסיף את "רשימת המשאלות" שלו, המחולקת לפי קטגוריות (אוכל,קניות,חוויות וחיות) ולפרסם אותה כפוסט לפייסבוק שלו. קיימת אפשרות להוסיף משאלה עם תמונה וללא תמונה. אפשר לסמן ברשימה כאשר השגת את המשאלה שלך, אפשר למחוק משאלה. כל הנתונים נשמרים עם המשתמש.   
  הוספנו progress bar שעוקב אחרי התקדמות המשתמש ברשימה שלו.
* אימונים**-** המשתמש יכול להוסיף את האימונים שהוא ביצע, בהוספת אימון יש להוסיף: סוג אימון,תאריך, משך זמן האימון וקלוריות שנשרפו. ישנה תצוגה בפרופיל של המשתמש בטבלה של כל האימונים שביצע.  
  המשתמש יכול לקבל סטטיסטיקות של האימונים שהוא ביצע- כמה אימונים הוא עשה כל חודש וכמה קלוריות הוא שרף כל חודש. כל הנתונים נשמרים עם המשתמש.  
  הוספנו חישובי סטטיסטיקות לפי פרמטרים נוספים.

### תבנית מס' 1 – Observer

* סיבת הבחירה / שימוש בתבנית:

מערכת ה Wishlist משתמשת בתבנית Observer כדי לעדכן באופן אוטומטי את כל הצופים (Observers) כאשר מתבצע שינוי ברשימת המשאלות)משאלה שבוצעה\לא בוצעה). הדבר מאפשר למעקב אחר התקדמות המשתמש באמצעות progress bar ולעדכון בזמן אמת של התצוגה.  
-**מודולריות**: אפשר להוסיף עוד צופים בלי לשנות את ה- WishlistManager.

**-עדכון אוטומטי**: כל שינוי מתעדכן מיידית בכל התצוגות הרלוונטיות.

**-הפרדה בין לוגיקה לתצוגה**: אין תלות בין ה-WishlistManager לבין הדרך שבה מוצגת ההתקדמות.

* אופן המימוש:
* **WishlistManager (Subject)-**

שומר רשימה של **Observers**

בכל שינוי ברשימה (הוספה/הסרה/עדכון), הוא מודיע לכל הצופים באמצעות NotifyObservers()

* **WishlistNotifier (Observer)-**

מקבל עדכון מWishlistManager כאשר כל הפריטים הושלמו.

מציג הודעה אם המשתמש השלים את כל המשאלות.

* **WishlistProgressTracker (Observer)-**

משתמש ב-ProgressBar כדי להציג באופן ויזואלי את ההתקדמות.

מחשב את אחוז ההתקדמות על בסיס מספר הפריטים שסומנו כבוצעו.

* מיקום בקוד:  
  מחלקות: WishlistNotifier, WishlistProgressTracker, IWishlistObserver

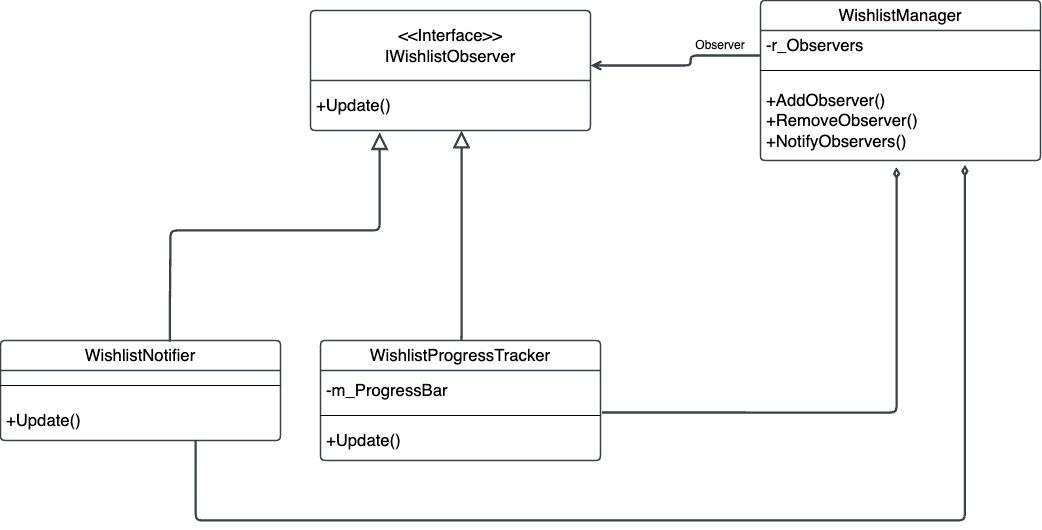
ב :WishlistManager  
שורות 93-108.  
שימוש ב -NotifyObservers 122,39,71,84,153

* Sequence Diagram

A diagram of a diagram

AI-generated content may be incorrect.

* Class Diagram



### תבנית מס' 2 – Strategy

* סיבת הבחירה / שימוש בתבנית:

במערכת הסטטיסטיקות, למשתמש יש מספר דרכים שונות לחישוב הנתונים (קלוריות, תדירות אימונים, משך זמן וכו').  
שימוש בתבניתStrategy מאפשר להחליף בקלות את האלגוריתם לחישוב הסטטיסטיקות מבלי לשנות את הקוד הראשי של המערכת.  
כך, ניתן להוסיף חישובים חדשים או לשנות את הקיימים מבלי להשפיע על שאר המערכת.  
-**גמישות:** ניתן להוסיף בקלות עוד אסטרטגיות חישוב מבלי לגעת בלוגיקה הראשית.

-**מודולריות:** כל חישוב מוגדר כיישות נפרדת, מה שמקל על תחזוקה והרחבה.

-**הפרדה בין נתונים ללוגיקה:** StatisticsForm לא צריך לדעת איך מתבצע החישוב, הוא פשוט מפעיל את m\_CurrentStrategy.

* אופן המימוש: (מימשנו באופן בו גיא לימד בכיתה)

**-IWorkoutStatisticStrategy**ממשק מגדיר את הפונקציה Calculate, שאותה כל אסטרטגיה מממשת באופן שונה.

**מימושים שונים של אסטרטגיות**:

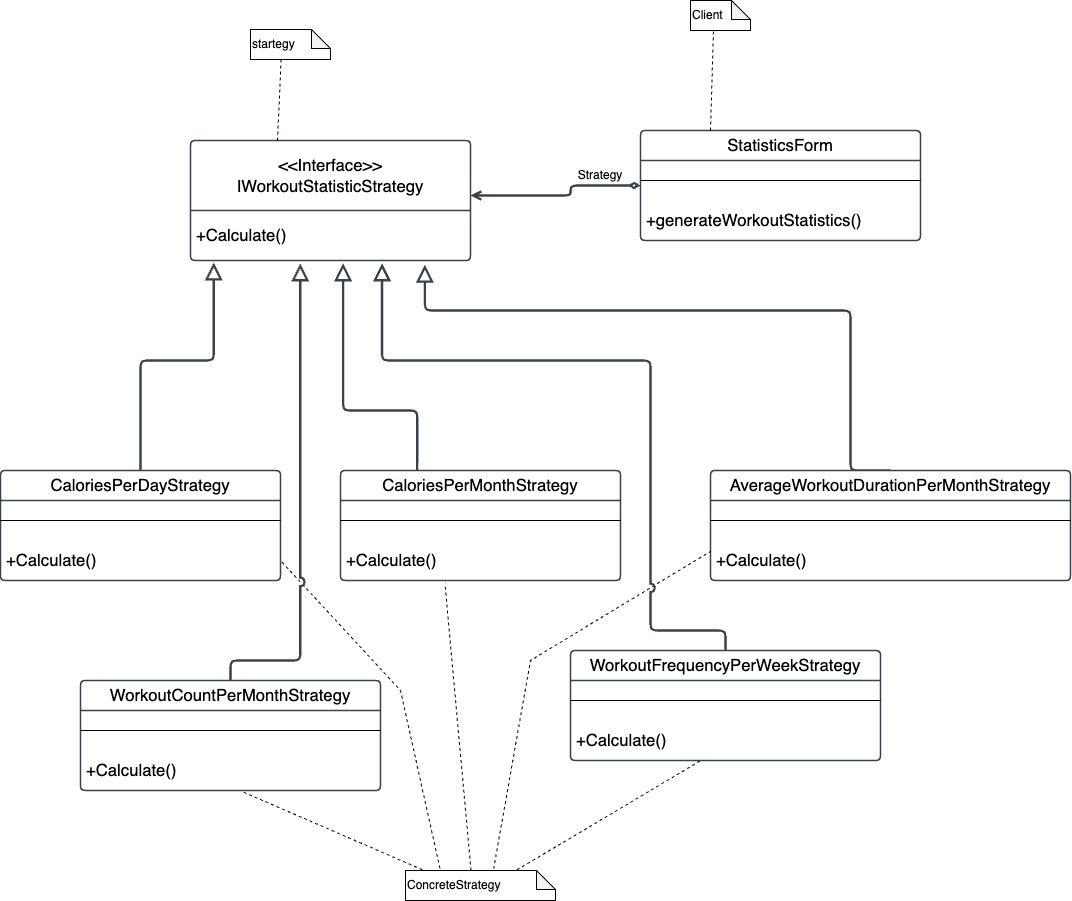
* CaloriesPerDayStrategy- מחשבת את סך הקלוריות שנשרפו בכל יום.
* CaloriesPerMonthStrategy- מחשבת את סך הקלוריות שנשרפו בכל חודש.
* WorkoutCountPerMonthStrategy- סופרת את כמות האימונים שבוצעו בכל חודש.
* WorkoutFrequencyPerWeekStrategy- מחלק את האימונים לפי שבועות ומחשב את התדירות השבועית.
* AverageWorkoutDurationPerMonthStrategy-מחשבת את משך האימון הממוצע לכל חודש.

**StatisticsForm** מחזיק משתנה m\_CurrentStrategy ומפעיל את החישובים בהתאם לבחירת המשתמש.

* מיקום בקוד:

מחלקות: CaloriesPerDayStrategy, CaloriesPerMonthStrategy, WorkoutCountPerMonthStrategy, WorkoutFrequencyPerWeekStrategy, AverageWorkoutDurationPerMonthStrategy,IWorkoutStatisticStrategy  
בStatisticsForm: שורות: 11, 24, וכל השאר מתבססות לפי המשתנה m\_CurrentStrategy

* A diagram of a diagram

  AI-generated content may be incorrect.Sequence Diagram
* Class Diagram

### תבנית מס' 3 – Iterator

* סיבת הבחירה / שימוש בתבנית:

נתוני פייסבוק (פוסטים, אלבומים, חברים וכו') מגיעים במבנים מורכבים ולא ניתנים לאיטרציה ישירה.  
תבנית Iterator מאפשרת לעבור על הנתונים בצורה אחידה, ללא תלות במבנה הפנימי שלהם.  
כך ניתן להציג נתונים בקלות בתצוגה ולהוסיף סוגי נתונים חדשים בלי לשנות את הלוגיקה הראשית

- **מעבר על נתונים בקלות:** foreach עובד ישירות על הנתונים ללא צורך בגישה ידנית לרשימה.

-**גמישות:** אפשר להחליף את המבנה הפנימי.

- **הפרדה בין אחסון לשליפה:**FacebookCollection<T> שומר נתונים, ו- FacebookEnumerator<T> אחראי על שליפתם.

* אופן המימוש:

**FacebookCollection<T>** - מחלקה המממשת IEnumerable<T> ומאפשרת איטרציה על נתוני פייסבוק.

**FacebookEnumerator<T>** - מחלקה המממשת IEnumerator<T> לצורך ניהול האיטרציה על הנתונים.

**FacebookManagerUI** - משתמש באובייקטים אלו כדי לעבור על הנתונים ולשייך אותם לתצוגה המתאימה.

* מיקום בקוד:  
  מחלקות: FacebookEnumerator, FacebookCollection

ב :FacebookManagerUIשורות 27-50

* Sequence Diagram

A diagram of a computer flowchart

AI-generated content may be incorrect.

* Class Diagram

A diagram of a computer

AI-generated content may be incorrect.